



JSP and Servlet Integration

Session 710





JSP and Servlet Integration

Karl Hsu
WebObjects Engineering

JSP and Servlets!

- Running WebObjects applications as servlets
- Accessing WebObjects components and actions from JSP pages



Assumptions

- You understand what a Servlet is
- You understand what JSPs are
- But in case you do not . . .

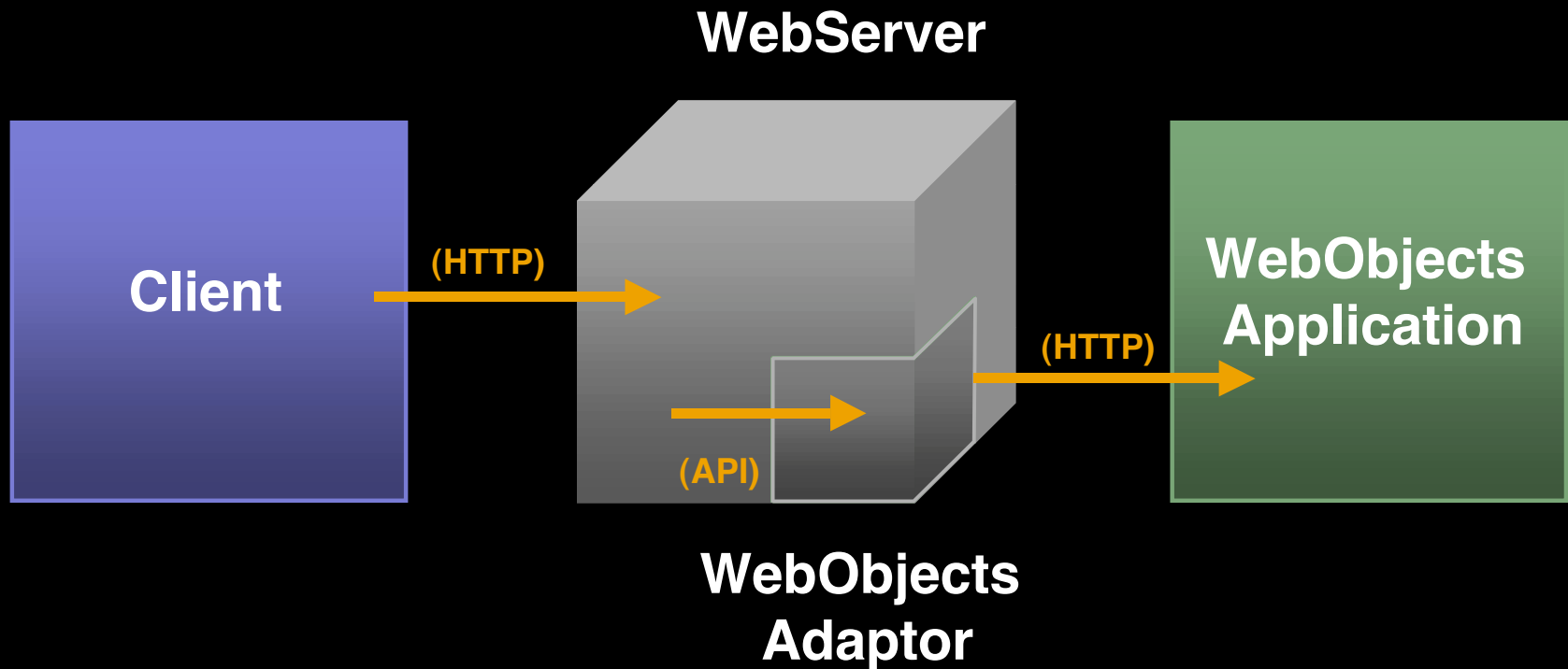


Servlets

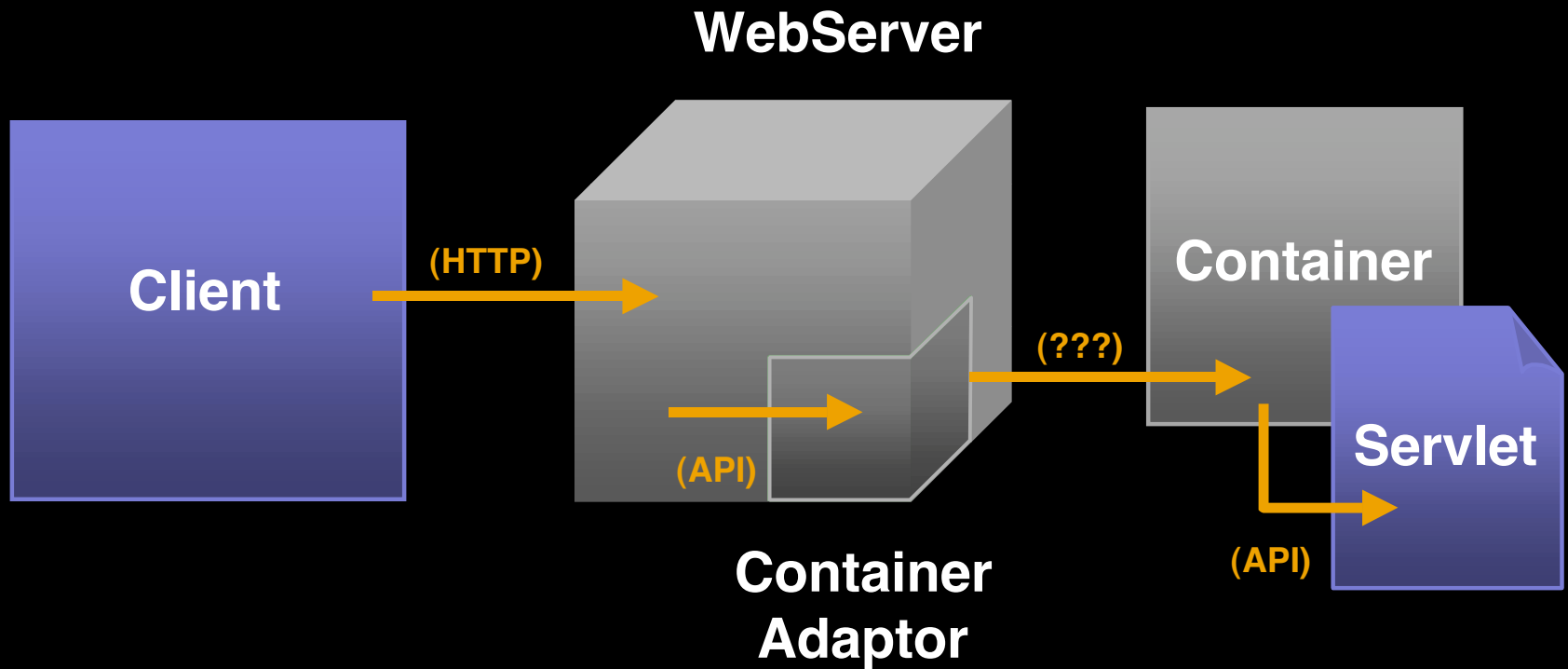
- Programmatic way of generating dynamic HTML
- Run inside of a Servlet Container
 - Tomcat
 - WebLogic
- You write a Servlet subclass which gets a request and a response object
 - You do all the work!



How Does WebObjects Work?



How Do Servlets Work??





On to WebObjects!

Servlet Integration Overview

- New to WebObjects 5.1!
- Allows a WebObjects application to run as a Servlet inside of a Servlet Container
- Lays the foundation for JSP integration
 - We will talk more about JSP integration in a while . . .



Why Did We Do This?

- So that you do not have to fight against Management or Clients
 - “We have standardized on WebLogic”
- Now you can satisfy everyone!
 - Treat WebObjects as the world’s most powerful Servlet library



How Did We Do This?

- Servlets are deployed in self-contained WAR files
- We actually run as a servlet inside a WAR file
 - We have a servlet subclass that takes the place of the `WODefaultAdaptor`
- Goal was to make it transparent to the WO developer and the non-WO deployer
- Servlet “adaptor” loads the rest of the WO classes at runtime
 - WO must be installed on the application server



What Can You Run?

- Regular WebObjects applications
- Direct To Web applications
- Java Client applications
 - Even Direct To JavaClient!



Where Can You Run?

- Tomcat
 - 3.x on MacOSX (Server)
 - 4.x on MacOSX (Server)
- WebLogic
 - 6.1 on Win2K and Solaris
- Other Servlet Containers?
 - WebSphere (soon)

**GOAL FOR
NEW RELEASE**



What About Deployment?

- Different from typical WebObjects deployment
- Monitor (and wotaskd) cannot be used to control WebObjects applications deployed as Servlets
 - Use the container's management tools
- WebObjects HTTP adaptors are not used either
 - Use the container's HTTP adaptor



Converting Existing Projects

1. Open the Project
2. Add the JavaWOJSPServlet framework
3. If a ProjectBuilderWO project, add the missing Makefile.preamble variables
4. That's all, folks . . .



New Projects

1. Create a new Project
2. When prompted, select “Deploy in a JSP/Servlet Container”
3. There is no step 3 . . .



Project Builder Details

What are these new files and folders?

- “Servlet Resources” subfolder added to Resources folder
 - Anything added to this folder will be added to the WAR
- web.xml.template file in “Servlet Resources”
 - Used to customize the web.xml (deployment descriptor) in the WAR
 - Placeholders surrounded by ‘%’ characters are replaced at compilation with the correct values



Project Builder Details

What are these new Build Settings?

- “SERVLET_WEBAPPS_DIR”
 - Directory where the WAR is created
- “SERVLET_COPY_JARS”
 - Should framework jars be copied into the WAR?
 - Only used for accessing classes directly from JSPs or Servlets



Project BuilderWO Details

What are these new files and folders?

- “JSP/Servlet WEB-INF” folder
 - Used to hold jars, classes and tlds
 - Items are auto-routed to the correct location in the WAR file
- “JSP/Servlet Resources” folder
 - Anything added to this folder will be added to the WAR
- Nested directories will retain their structure, but must be edited outside of PBWO



Project BuilderWO Details

What are these new makefile variables?

- Added to the Makefile.preamble
- “SERVLET_APP_MODE”
 - Set to “Deployment” or “Development”
 - Are images loaded from the WebServer or the app?
- “SERVLET_WEBAPPS_DIR”
 - Directory where the WAR is created
- “SERVLET_COPY_JARS”
 - Should framework jars be copied into the WAR?
 - Only for accessing WebObjects classes directly from JSPs or Servlets



How Do I Deploy?

- Build your project
 - WAR file should be created in the `SERVLET_WEBAPPS_DIR`
 - Defaults to the same location as the `.woa` if no value specified
- Use the container's tools to deploy the application
 - Many containers have a setting for automatic deployment of applications in a “special” directory
- URL

`http://host:<port>/MyApp/WebObjects/MyApp.woa`



Deployment Quibbles

- Remember how the Servlet “adaptor” loads the WO classes at runtime?
 - It needs to know where to find the classes
- Three variables define locations of the classes
 - Variables are defined in the web.xml to work for the development environment
 - Can be overridden by System Properties set by the container



Which Three Variables?!

- `WOAINSTALLROOT`
 - Directory where your woa is
- `LOCALROOT`
 - Directory where your frameworks are
- `WOROOT`
 - Directory where the System frameworks are



Further Discussion

- WOSessions are “piggybacked” on top of JSP/Servlet sessions
 - If you subclass or modify the SessionStore, there might be issues
- Some WO properties are not used
- System Properties are shared between all apps
- WODefaultAdaptor is not used at all
- We have a custom WOContext subclass
 - If you subclassed WOContext, subclass WOServletContext instead





Demo

Megan Moore
WebObjects Engineering

JSPs

- Java Server Pages
- Template-based way of generating dynamic HTML
- Compiled into Servlets at runtime by Servlet Containers
- You write HTML with bits of embedded Java Code
 - You can also call out to other servlets or include other JSP pages



JSP Integration Overview

- Also new to WebObjects 5.1!
- Based on (and requires) Servlet integration
 - You must run as a Servlet in a Container!
- Use WOComponents and Direct Actions to generate HTML for your JSP pages



Why Did We Do This?

- Allows you to write mixed JSP and WebObjects applications
- From a JSP, use WebObjects to do complex things easily
 - Custom WOComponents
 - Complex, template-based HTML
 - WORepetitions, WOProgressBar, etc.
 - Data Persistence
 - Enterprise Objects instead of JDBC



How Did We Do This?

- Taglibs are custom JSP tags

```
<%@ taglib uri="/tutorial-template" prefix="tt" %>  
<tt:tag>  
  body  
</tt:tag>
```

- We wrote a custom tag library (taglib)
- Defined in a tld file
 - WOTaglib_1_0.tld inside
JavaWOJSPServlet.framework



Using WOComponents 1

"component" tag

- Attributes
 - className: Name of the component (required)
 - bodyContentOnly: Should return everything, or only content between <body> tags (optional, default true)
 - mergeResponseHeaders: Should merge WOResponse headers into ServletResponse (optional, default false)



Using WOComponents 2

"component" tag

- “extraHeader” sub tag
 - Defines a header to be passed to the component
 - “key” attribute: the header key
 - “value” attribute: the header value
- “binding” sub tag
 - Defines a binding to be passed to the component
 - “key” attribute: the binding key
 - “value” attribute: the binding value



WOComponent Code

```
<%@ taglib uri="/WOTaglib" prefix="wo"%>
<%
int val = (new Random()).nextInt(10) + 1;
String value = Integer.toString(val);
%>
```

```
<wo:component className="WOCompletionBar">
  <wo:binding key="align" value='<%= "RIGHT" %>' />
  <wo:binding key="numberformat" value='<%= "0" %>' />
  <wo:binding key="value" value='<%= value %>' />
  <wo:binding key="valueMax" value='<%= new Integer(10)%>' />
  <wo:binding key="valueMin" value='<%= "1" %>' />
  <wo:binding key="width" value='<%= "100" %>' />
</wo:component>
```



Using Direct Actions 1

"directAction" tag

- Attributes
 - `actionName`: Name of the direct action (required)
 - `className`: Name of the direct action class (optional)
 - `bodyContentOnly`: Should return everything, or only content between `<body>` tags (optional, default true)
 - `mergeResponseHeaders`: Should merge `WOResponse` headers into `ServletResponse` (optional, default false)



Using Direct Actions 2

"directAction" tag

- “extraHeader” sub tag
 - Defines a header to be passed to the direct action
 - “key” attribute: the header key
 - “value” attribute: the header value
- “formValue” sub tag
 - Defines a form value to be passed to the direct action
 - “key” attribute: the form key
 - “value” attribute: the form value



Direct Action Code

```
<%@ taglib uri="/WOTaglib" prefix="wo" %>
```

```
<wo:directAction actionName="MyAction"  
  bodyContentOnly="false">
```

```
  <wo:formValue key="fKey1" value="fValue1" />
```

```
  <wo:formValue key="fKey2" value="fValue2" />
```

```
  <wo:extraHeader key="hKey1" value="hValue1" />
```

```
</wo:directAction>
```



Anything Else?

- Remember `SERVLET_COPY_JARS`?
- It allows you to directly reference WebObjects classes from JSPs or Servlets without using the tag libraries
- You must make sure that WebObjects is initialized first
 - This happens automatically if you use the taglib
`<% WOServletAdaptor.initStatics(application); %>`



Direct References Code

```
<%@ taglib uri="/WOTaglib_1_0.tld" prefix="wo" %>
<%@ page import = "com.webobjects.eoaccess.*" %>
<%@ page import = "com.webobjects.eocontrol.*" %>
<%@ page import = "com.webobjects.foundation.*" %>
<%@ page import = "com.webobjects.jspervlet.*" %>
<%
WOServletAdaptor.initStatics(application);

EOFetchSpecification fs = new
    EOFetchSpecification("Movie",null,null);
EOEditingContext ec = new EOEditingContext();
NSArray objectArray = ec.objectsWithFetchSpecification(fs);
%>
<wo:component className="MyWORepComponent">
    <wo:binding key="objectArray" value='<%= objectArray %>' />
</wo:component>
```





Demo

Megan Moore
WebObjects Engineering

What About the Future?

- Servlet Single Directory Deployment
 - Instead of a WAR file + WO installation
 - Single Directory that is a standalone application (nothing else needs to be installed!)
 - No need to set System Properties or modify the web.xml file
 - Controlled by a new Build Setting/Makefile variable—SERVLET_SINGLE_DIR_DEPLOY



WebObjects Beta

- To be considered for the beta
Appleseed.apple.com/webobjects



WebObjects Lab

- Located downstairs in Room L
- Lab hours
 - Monday 12:00pm–6:00pm
 - Tuesday 9:00am–2:00pm
 - Wednesday 9:00am–6:00pm
 - Thursday 9:00am–6:00pm
 - Friday 9:00am–6:00pm



Roadmap

708 Enterprise Java Beans

Room A1
Thurs., 9:00am

**709 Deploying WebObjects
Applications**

Room A1
Thurs., 10:30am

FF013 WebObjects

Room A1
Fri., 3:30pm



Who to Contact

Toni Trujillo Vian

Director, WebObjects Engineering

webobjects@apple.com

Bob Fraser

WebObjects Product Manager

webobjects@apple.com

Apple Professional Services (Training, Support, Consulting)

(800) 848-6398

services@apple.com



For More Information

- WebObjects Developer Documentation
<http://developer.apple.com/techpubs/webobjects>
- Apple Professional Services Technical Support
www.apple.com/services/technicalsupport
- Other places
www.apple.com/webobjects
developer.apple.com/webobjects
www.apple.com/services
www.info.apple.com/webobjects
- Subscribe to
webobjects-announce@apple.com



Documentation



- Developing Applications Using JavaServer Pages and Servlets
- Discovering WebObjects for HTML



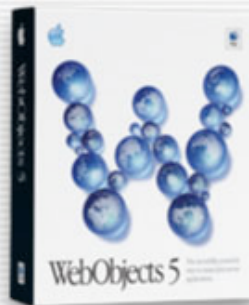
How to Access Documentation

- Most up-to-date: PDF and HTML
<http://developer.apple.com/techpubs/webobjects>
- Hardcopy print-on-demand
Vervante.com under **Related Resources**
- Product CD
Documents folder and installed in
/Developer/Documentation/WebObjects
- In the box (localized)
Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications, Discovering WebObjects for HTML
- Check ADC News for latest updates
<http://developer.apple.com/devnews>





Q&A



Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**